



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/705,557	11/02/2000	Paul F. Klein	G&C30695.16-US-U1	8702

22462 7590 05/07/2004

GATES & COOPER LLP  
HOWARD HUGHES CENTER  
6701 CENTER DRIVE WEST, SUITE 1050  
LOS ANGELES, CA 90045

EXAMINER

BLACKWELL, JAMES H

ART UNIT	PAPER NUMBER
----------	--------------

2176

3

DATE MAILED: 05/07/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/705,557

Applicant(s)

KLEIN ET AL.

Examiner

James H Blackwell

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 02 November 2000.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-57 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-57 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 02 November 2000 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. §§ 119 and 120

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).  
\* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.  
a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

## Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_

### DETAILED ACTION

1. Claim 8 is objected to because of the following informalities: Claim 8 depends on itself. For purposes of examination, Examiner has assumed that Claim 8 depends on Claim 1. Appropriate correction is required.

### ***Claim Rejections - 35 USC § 102***

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-4, 8-16, 19-23, 27-35, 37-39, 40-42, 47-54, and 56-57 are rejected under 35 U.S.C. 102(e) as being anticipated by Landsman et al. (hereinafter Landsman, U.S. Patent No. 6,314,451).

In regard to independent Claim 1 (and similarly independent Claims 20, and 39), Landsman teaches embedding an "*advertising tag*" into a referring page. This tag contains two components. One component effectively downloads, from a distribution HTTP (web) server and to an extent necessary, and then persistently instantiates an agent, implemented as a "*light-weight*" Java applet, at the client browser. This agent then "*politely*" and transparently downloads advertising files (media and, where

necessary, player files), originating from an ad management system residing on a third-party advertising HTTP (web) server, for a given advertisement into browser disk cache (also in the case of media files into the browser RAM cache) and subsequently plays those media files through the browser on an interstitial basis and in response to a user click-stream (Col. 9, line 67; Col. 10, lines 1-12; compare to Claim 1 (and similarly Claims 20, and 39), “*(a) requesting a web page; (b) receiving a web page in a browser in response to the request, wherein the web page comprises an applet tag; (c) requesting an applet identified by the applet tag; (d) receiving the applet; (e) executing the applet,...*”). Landsman also teaches that in response to this request for an AdDescriptor file, ad management system (25) then selects a specific advertisement to be delivered to client PC (5). This selection can be selected on a predefined or random basis, or based on user preference or other user-specific information previously collected from and associated with the user then operating browser (7). Such user-specific information, such as prior buying patterns, could have been appropriately pre-collected at the client PC, previously uploaded to ad management system (25) and processed there such that, upon receipt of the AdDescriptor request, system (25) would then select and download an appropriate advertisement specifically targeted to the user then situated at the client PC. In any event, once system (25) selects the advertisement, through whatever selection metric it employs, the corresponding AdDescriptor file is then downloaded, as symbolized by line (66), to agent server (15) (here being a proxy server) which, in turn, as symbolized by line (68), supplies that file to the AdController agent then executing under web browser

(7) (Col. 21, lines 13-31; compare to Claim 1 (and similarly Claims 20, and 39), “... ***requesting a web object that is likely to be accessed next***”). Landsman also teaches that once the AdDescriptor file is downloaded to the client PC, via agent server (15), the AdController then “*politely*” downloads, as symbolized by block (70) shown in Figs. 1B and 1C, into the browser disk cache each media and player file, as specified in the AdDescriptor file--to the extent that file does not already reside on the hard disk of the client PC (Col. 22, lines 4-9; Compare to Claim 1 (similarly Claims 20, and 39), “... ***receiving the requested web object; and copying the requested web object into a cache of the browser***”).

In regard to dependent Claim 2 (and similarly dependent Claims 21, and 40), Landsman teaches an HTML advertising tag is embedded into a referring web page (see Abstract; compare with Claim 2 (and similarly Claims 21, and 40), “... ***the web object is a hypertext markup language (HTML) document***”).

In regard to dependent Claim 3 (and similarly dependent Claims 22, and 41), Landsman teaches that for the selected advertisement, the AdDescriptor file is a text file that contains a manifest, i.e., a list, of file names and corresponding network locations (URLs) at which these files reside, and player instructions and configuration parameter values necessary to play the entire advertisement through web browser (7) to the user. Fig. 20 shows contents of typical AdDescriptor file (2000) for a PointCast Java advertisement. Specifically and as shown in section 4C of file (2000), this AdDescriptor file lists file names with partial addresses on the ad management system of all media files that constitute content for that advertisement, and, in section 1 of this file, all Java

player files necessary to play all the media files. This file also respectively specifies, here shown in sections 3 and 4B, an order in which the various media files are to be played, and various configuration parameters needed to properly configure the operation of each player to play each corresponding media file (Col. 21, lines 32-48; compare to Claim 3 (and similarly Claims 22, and 41), ***“... the applet is further configured to receive an object list of one or more web objects likely to be accessed next and wherein the request for the web object is for a web object in the object list”***).

In regard to dependent Claim 4 (and similarly dependent Claims 23, and 42), Landsman teaches ad management system (25) then selects a specific advertisement to be delivered to client PC (5). This selection can be selected on a predefined or random basis, or based on user preference or other user-specific information previously collected from and associated with the user then operating browser (7). Such user-specific information, such as prior buying patterns, could have been appropriately pre-collected at the client PC, previously uploaded to ad management system (25) and processed there such that, upon receipt of the AdDescriptor request, system (25) would then select and download an appropriate advertisement specifically targeted to the user then situated at the client PC (Col. 21, lines 13-25; compare to Claim 4 (and similarly Claims 23, and 42), ***“... the one or more web objects in the object list are ordered by statistical significance and the applet requests each web object in the list in the statistical significance order”***).

In regard to dependent Claim 8, Landsman teaches that through so-called "polite" downloading, media and player files are downloaded to browser (7) during browser idle time intervals, with the downloading suspended during each ensuing interstitial interval after the user instructs browser (7) to navigate to a new content web page. In this manner, while a fully downloaded advertisement is interstitially played from browser cache, the new content page is downloaded over the full bandwidth of communications link (9). Advantageously, the communications link is freed during each interstitial interval to just carry web page content, thereby expediting download of content pages. If, due to the occurrence of an interstitial interval, the AdController applet suspends downloading of an advertisement file, then upon termination of this interval, this applet then resumes downloading at a location in that file at which downloading had stopped, thus conserving communication bandwidth and reducing download time (Col. 22, lines 10-26; compare to Claim 8, "*... the applet does not interfere with normal processing of the browser*").

In regard to independent Claim 9 (and similarly independent Claims 28, and 47), Landsman teaches embedding an "*advertising tag*" into a referring page. This tag contains two components. One component effectively downloads, from a distribution HTTP (web) server and to an extent necessary, and then persistently instantiates an agent, implemented as a "*light-weight*" Java applet, at the client browser. This agent then "*politely*" and transparently downloads advertising files (media and, where necessary, player files), originating from an ad management system residing on a third-party advertising HTTP (web) server, for a given advertisement into browser disk cache

(also in the case of media files into the browser RAM cache) and subsequently plays those media files through the browser on an interstitial basis and in response to a user click-stream (Col. 9, line 67; Col. 10, lines 1-12; compare to Claim 9 (and similarly Claims 28, and 47), “... **(b) an application on the server, the application configured to: (1) receive a request for a web page; (2) obtain a web page comprising an applet tag; (3) transmit the web page to a client; (4) receive a request for an applet identified by the applet tag; and (5) transmit the applet to the client ...**”).

Landsman also teaches that in response to this request for an AdDescriptor file, ad management system (25) then selects a specific advertisement to be delivered to client PC (5). This selection can be selected on a predefined or random basis, or based on user preference or other user-specific information previously collected from and associated with the user then operating browser (7). Such user-specific information, such as prior buying patterns, could have been appropriately pre-collected at the client PC, previously uploaded to ad management system (25) and processed there such that, upon receipt of the AdDescriptor request, system (25) would then select and download an appropriate advertisement specifically targeted to the user then situated at the client PC. In any event, once system (25) selects the advertisement, through whatever selection metric it employs, the corresponding AdDescriptor file is then downloaded, as symbolized by line (66), to agent server (15) (here being a proxy server) which, in turn, as symbolized by line (68), supplies that file to the AdController agent then executing under web browser (7) (Col. 21, lines 13-31; compare to Claim 9 (and similarly Claims 28, and 47), “... **wherein the applet is configured to: (i) request, from the server, a**



***web object that is likely to be accessed next; and (ii) copy the requested web object into a cache of a browser on the client; (6) transmit the requested web object to the client”).***

In regard to dependent Claim 10 (and similarly dependent Claims 29, and 48), Landsman teaches that the advertising tag is itself embedded in a content web page. The advertising tag, as one of its components, references a JavaScript file (which contains a "script") stored on a distribution server (Col. 11, lines 40-43; compare to Claim 10 (and similarly Claims 29, and 48), ***“... the obtaining comprises receiving a static web page from a web page library on the server”***).

In regard to dependent Claim 11 (and similarly dependent Claims 30, and 49), Landsman teaches an advertising tag which downloads a JavaScript file from an agent server. This file, in turn, is then interpreted and executed as a script, by the browser. Applet tags are dynamically written by the script into the referring web page in lieu of advertising tag so as to form a modified web page (Col. 17, lines 57-67; compare to Claim 11 (and similarly Claims 30, and 49), ***“... adding an applet tag to the static web page”***).

In regard to dependent Claim 12 (and similarly dependent Claims 31, and 50), Landsman teaches an advertising tag which downloads a JavaScript file from an agent server. This file, in turn, is then interpreted and executed as a script, by the browser. Applet tags are dynamically written by the script into the referring web page in lieu of advertising tag so as to form a modified web page (Col. 17, lines 57-67; compare to Claim 12 (and similarly Claims 31, and 50), ***“... the obtaining comprises using a filter***

*that dynamically tags the web page as the web page is being transmitted to the client”).*

In regard to dependent Claim 13 (and similarly dependent Claims 32, and 51), Landsman teaches an advertising tag which downloads a JavaScript file from an agent server. This file, in turn, is then interpreted and executed as a script, by the browser. Applet tags are dynamically written by the script into the referring web page in lieu of advertising tag so as to form a modified web page (Col. 17, lines 57-67; compare to Claim 13 (and similarly Claims 32, and 51), “... *the obtaining comprises dynamically creating a web page*”).

In regard to dependent Claim 14 (and similarly dependent Claims 33, and 52), Landsman teaches that for the selected advertisement, the AdDescriptor file is a text file that contains a manifest, i.e., a list, of file names and corresponding network locations (URLs) at which these files reside, and player instructions and configuration parameter values necessary to play the entire advertisement through web browser (7) to the user. Fig. 20 shows contents of typical AdDescriptor file (2000) for a PointCast Java advertisement. Specifically and as shown in section 4C of file (2000), this AdDescriptor file lists file names with partial addresses on the ad management system of all media files that constitute content for that advertisement, and, in section 1 of this file, all Java player files necessary to play all the media files. This file also respectively specifies, here shown in sections 3 and 4B, an order in which the various media files are to be played, and various configuration parameters needed to properly configure the operation of each player to play each corresponding media file (Col. 21, lines 32-48;

compare to Claim 14 (and similarly Claims 33, and 52), “... *transmitting a web object list to the client*”).

In regard to dependent Claim 15 (and similarly dependent Claims 34, and 53), Landsman teaches that in sharp contrast to conventional server-based accounting of web advertisements, our inventive technique provides highly accurate client-side accounting of each user impression. Each log entry, produced by the AdController applet, specifies a successful presentation of a complete advertisement at a client browser. This entry may include a source of the ad content, i.e., in terms of the URL of the associated ad management system, a title of the advertisement and the URL of the referring web page. Other client-side information can be measured and included in each entry, such as: an amount of time during which the advertisement was rendered by the browser (presumably during which the user dwelled on the advertisement); as well as an identification, in terms of a URL, of a content web page to which the user next navigated (particularly if the user reached that page through a hotlink displayed in the advertisement). Subsequently, the AdController applet uploads the log entries to the advertising server. These entries will be collectively processed, as needed, to permit shared ad revenues from web-based advertisers to be properly allocated among different web page content providers (Col. 13, lines 34-54; compare to Claim 15 (and similarly Claims 34, and 53), “... *maintaining access statistics for the web page, wherein the access statistics are statistics for web objects accessed after the web page*”).

In regard to dependent Claim 16 (and similarly dependent Claims 35, and 54), Landsman teaches that in sharp contrast to conventional server-based accounting of web advertisements, our inventive technique provides highly accurate client-side accounting of each user impression. Each log entry, produced by the AdController applet, specifies a successful presentation of a complete advertisement at a client browser. This entry may include a source of the ad content, i.e., in terms of the URL of the associated ad management system, a title of the advertisement and the URL of the referring web page. Other client-side information can be measured and included in each entry, such as: an amount of time during which the advertisement was rendered by the browser (presumably during which the user dwelled on the advertisement); as well as an identification, in terms of a URL, of a content web page to which the user next navigated (particularly if the user reached that page through a hotlink displayed in the advertisement). Subsequently, the AdController applet uploads the log entries to the advertising server. These entries will be collectively processed, as needed, to permit shared ad revenues from web-based advertisers to be properly allocated among different web page content providers (Col. 13, lines 34-54; compare to Claim 16 (and similarly Claims 35, and 54), *"... the maintaining comprises maintaining a web agent table with slots, wherein each slot represents a location to find the access statistics for a web page"*).

In regard to dependent Claim 27, Landsman teaches that through so-called "polite" downloading, media and player files are downloaded to browser (7) during browser idle time intervals, with the downloading suspended during each ensuing

interstitial interval after the user instructs browser (7) to navigate to a new content web page. In this manner, while a fully downloaded advertisement is interstitially played from browser cache, the new content page is downloaded over the full bandwidth of communications link (9). Advantageously, the communications link is freed during each interstitial interval to just carry web page content, thereby expediting download of content pages. If, due to the occurrence of an interstitial interval, the AdController applet suspends downloading of an advertisement file, then upon termination of this interval, this applet then resumes downloading at a location in that file at which downloading had stopped, thus conserving communication bandwidth and reducing download time (Col. 22, lines 10-26; compare to Claim 27, "... *the applet does not interfere with normal processing of the browser*").

***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 5, 24, and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Landsman in view of O'Brien et al. (hereinafter O'Brien, U.S. Patent No. 6,055,569).

In regard to dependent Claim 5 (and similarly dependent Claims 24, and 43), Landsman fails to specifically teach that *the request for a web object comprises a request for statistical information regarding a web page most likely to be accessed directly after the current web page being viewed on the browser and wherein only a statistically significant web object is received by the applet*. However, O'Brien teaches that a smart browser working in conjunction with a HTTP server that selectively downloads WWW pages into the browser's memory cache. The determination of which pages to download is a function of a probability weight assigned to each link on a Web page. By evaluating that weight to a predetermined browser criteria, only those pages most probably to be downloaded are stored in the browser's memory cache. The download is done in the background while the browser user is viewing the current Web page on the monitor. This greatly enhances the speed with which the viewer can "cruise" the Web while at the same time conserving system resources by not requiring the system to download all the possible links (see Abstract; compare to Claim 5 (and similarly Claims 24, and 43), "... *the request for a web object comprises a request for statistical information regarding a web page most likely to be accessed directly after the current web page being viewed on the browser and wherein only a statistically significant web object is received by the applet*"). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Landsman and O'Brien providing the benefit of accelerating web access by predicting user action.

Claims 6-7, 25-26, and 44-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Landsman in view of Borman et al. (hereinafter Borman, U.S. Patent No. 6,226,655).

In regard to dependent Claim 6 (and similarly dependent Claims 25, and 44), Landsman fails to teach *parse contents of the web page; create a web object list that contains each link to another web page that is identified from the parsing, wherein each web object requested by the applet is a web object from the web object list*. However, Borman teaches that the jumper is implemented as an application, such as an applet, which is sent to the browser by the search engine (Col. 12, lines 34-36; compare with Claim 6 (and similarly Claims 25, and 44), “... *the applet is further configured to*”). Borman also teaches that a first file of information is received which may include a first mark-up language to identify contents of the information, which contents include site identifiers. The site identifiers corresponding for example to file locations on the Internet. The first file is displayed in a browser window. Responsive to receiving the first file of information by the browser, the first file of information is parsed by a jumper to generate a list of site identifiers. This list of site identifiers is then stored by the jumper and displayed in a jumper window. Responsive to an activation by the user, a computer is directed to determine which of the stored site identifiers is currently selected and automatically selects an other. The other includes the first, the prior, the next, or the last on the list (Col. 3, lines 9-22; compare to Claim 6 (and similarly Claims 25, and 44), “... *parse contents of the web page; create a web object list that contains each link to another web page that is identified from the parsing, wherein each web object*

*requested by the applet is a web object from the web object list").* It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Landsman and Borman providing the benefit of generating a list of site identifiers (URLs).

In regard to dependent Claim 7 (and similarly dependent Claims 26, and 45), Landsman fails to teach that *upon receiving a web object requested from the web object list, the applet is further configured to: parse contents of the web object received; identify any link to another web page; add the identified link to the web object list.* However, Borman teaches that responsive to receiving the first file of information, the jumper parses the first file and extracts and stores a list comprised of first file site identifiers. The stored list of site identifiers is then displayed in the jumper window (Col. 3, lines 14-18; compare to Claim 7 (and similarly Claims 26, and 45), "... *upon receiving a web object requested from the web object list, the applet is further configured to: parse contents of the web object received; identify any link to another web page; add the identified link to the web object list*"). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Landsman and Borman providing the benefit of having identified links in a web object and creating a list of them thereby assisting with simplified Internet navigation.

In regard to dependent Claim 46, Landsman teaches that through so-called "polite" downloading, media and player files are downloaded to browser (7) during browser idle time intervals, with the downloading suspended during each ensuing



interstitial interval after the user instructs browser (7) to navigate to a new content web page. In this manner, while a fully downloaded advertisement is interstitially played from browser cache, the new content page is downloaded over the full bandwidth of communications link (9). Advantageously, the communications link is freed during each interstitial interval to just carry web page content, thereby expediting download of content pages. If, due to the occurrence of an interstitial interval, the AdController applet suspends downloading of an advertisement file, then upon termination of this interval, this applet then resumes downloading at a location in that file at which downloading had stopped, thus conserving communication bandwidth and reducing download time (Col. 22, lines 10-26; compare to Claim 46, “... ***the applet does not interfere with normal processing of the browser***”).

Claims 17, 36 and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Landsman in view of Batchelder et al. (hereinafter Batchelder, U.S. Patent No. 6,351,767).

In regard to dependent Claim 17 (and similarly dependent Claims 36, and 55), Landsman fails to teach *a hash function performed on a uniform resource locator (URL) for a web page identifies the slot containing the access statistics for that web page*. However, Batchelder teaches that When server (100) receives a URL from a client, the HTTP server (206) passes the URL to the URL Parser (303), which breaks the URL into different parts (Col. 7, lines 11-15; compare to Claim 17 (and similarly Claims 36, and 55), “... ***a hash function performed on a uniform resource locator (URL) for a web***”).

*page identifies the slot containing the access statistics for that web page").*

Batchelder fails to specifically teach *a hash function or identifies the slot containing the access statistics for that web page*. However, it would have been obvious to one of ordinary skill in the art at the time of invention to have modified the teaching of Batchelder to use a hash function on a URL because such a function would have provided a means to store the components of a URL in separate bins thereby assisting in isolating the path to the access statistics for a particular page.

Claims 18-19, 37-38, and 56-57 are rejected under 35 U.S.C. 103(a) as being unpatentable over Landsman.

In regard to dependent Claim 18 (and similarly dependent Claims 37, and 56), Landsman fails to teach *the access statistics in each slot are ordered by most linked to web objects to least linked to web objects*. However, it would have been obvious to one of ordinary skill in the art at the time of invention to have ordered access statistics in such a way because it would have made them easier to read and interpret by a user providing the benefit of quickly determining the most popular web objects.

In regard to dependent Claim 19 (and similarly dependent Claims 38, and 57), Landsman fails to teach *transmitting a subset of the slot for the requested web page to the applet*. However, it would have been obvious to one of ordinary skill in the art at the time of invention to have transmitted a subset of the slot to the applet providing the benefit of using the slot information to determine the most popular web objects.


***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to James H Blackwell whose telephone number is 703-305-0940. The examiner can normally be reached on Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph H Feild can be reached on 703-305-9792. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

James H. Blackwell  
04/22/04

  
JOSEPH FEILD  
SUPERVISORY PATENT EXAMINER